Trustworthy Data Sharing

Reimagining protection in the digital ecosystem

February 2025

UNSW SYDNEY | UTS Trustworthy Digital Society

1 Acknowledgements

This research was conducted in collaboration with the Trustworthy Digital Society. The authors gratefully acknowledge the support, resources, and intellectual contributions provided by all participating organisations and individuals in this collaborative effort to advance secure and ethical data sharing frameworks.

2 Executive Summary

The digital transformation has made data collection, use, and sharing increasingly ubiquitous, creating tremendous opportunities for individual, commercial, and public benefit. As AI continues to unlock the potential of unstructured data across text, images, sound, and video, we face unprecedented opportunities to create value through data sharing and innovative data products. However, the opportunities in data bring significant challenges that our current approaches are ill-equipped to address.

The current data sharing paradigm leaves users burdened with protection responsibilities and vulnerable as a result. Fragmented governance, siloed identity systems, inconsistent data models, and inefficient verification processes force users to navigate complex security decisions with limited support. In academic credential sharing, for example, students must manually manage sensitive documents while employers struggle to verify authenticity—creating significant risks of fraud, privacy breaches, and identity theft that ultimately fall on individual users to mitigate.

Our proposed new paradigm fundamentally reimagines data sharing by transferring protection responsibility from users to the system itself. Through unified governance frameworks, machine-readable policies, and metadata that accumulates throughout a data product's lifecycle, the system—not the user—evaluates risks and implements appropriate safeguards. This approach delivers significant benefits: automated verification, enhanced security, consistent policy enforcement, and improved user control. By shifting responsibility from individuals to the system, we can establish a more trustworthy, efficient, and secure data ecosystem that balances innovation with protection, ultimately enabling data sharing to reach its full potential while safeguarding all stakeholders.

3 The Current Paradigm

3.1 Overview: Our system is due for a change

The current paradigm of data sharing is characterised by several key practices and processes that define how data is managed and protected within organisations.

While providing foundational support for data sharing, current practices rely heavily on manual interpretations of policy, I and do not embed persistent controls or efficient governance.

The Australian Privacy Principles (APPs) provide a structured framework for managing personal information in Australia. These principles are part of the Privacy Act 1988 and apply to most Australian government agencies and private sector organisations.

The APPs require entities to manage personal information transparently, which includes having a clear and up-to-date privacy policy. APP entities must also provide individuals with the option of not identifying themselves or using a pseudonym when interacting with them, where practicable and lawful.

The principles guide the collection, use and disclosure of personal information. APP entities are required to handle personal information appropriately, destroying or de-identifying it when no longer needed. They must notify individuals about the purpose of the collection, how the information will be used, and to whom it may be disclosed.

Specific conditions must be met for using personal information and APP entities must take reasonable steps to ensure that any disclosure of personal information complies with the APPs, yet there is no way at present to mechanise these legal obligations.

When data is shared between two separate entities, they coordinate privacy compliance through several typical practices. First, they should establish formal data sharing agreements that outline the terms and conditions for data sharing, including privacy and security requirements, responsibilities, and procedures for handling data breaches.

Both entities must ensure their privacy policies align with the APPs and often a range of other requirements relevant to one or both organisations (e.g. GDPR), and outline how personal information will be managed, used, and protected. They should also implement controls to ensure that only authorised personnel can access the shared data, using secure methods for data transfer and storage.

Additionally, entities should conduct regular audits and monitoring to ensure compliance with privacy policies and data sharing agreements, helping to identify and address any potential issues promptly.

Both entities should also provide ongoing training and awareness programs for their staff to ensure they understand their privacy obligations and the importance of protecting personal information. By following these practices, entities can better coordinate privacy compliance and protect personal information when sharing data.

In summary, the current paradigm of data sharing can be described by the following four domains:

1. **Governance and policy:** The current paradigm is characterised by the Privacy Act 1988 and the Australian Privacy Principles, lacking specific technical standards with inconsistent interpretation across organisations.
2. **Identity:** Identity management is fragmented with multiple service platforms and logins, limited interoperability, and no universally accepted digital identity across ecosystems.

3. **Data modelling:** Data modelling is often inconsistent, leading to inefficient data exchanges, lacking common standards, leading to difficulties in machine-to-machine data exchange and slow integration.
4. **Integration and interoperability:** Integration is ad hoc, unstable, and inefficient, hence not scalable systems with custom agreements and manual processes that are not universally adopted or future-proof.

3.2 Shortcomings

The current paradigm of data sharing has several significant shortcomings that impact data privacy, security, customer experience, economic opportunity, satisfaction and overall digital adoption.

One major issue is that controls must be manually implemented on independent systems. Because controls often operate locally, they can be deliberately or accidentally removed from the control environment.

Reliance on data operators to interpret policies and implement the necessary controls leads to varying enforcement. Different operators may understand and implement policies in different ways, increasing the risk of privacy non-compliance and inadequate data protection measures. Additionally, conservative interpretations meaning that legal and constructive processing of data are not done.

This is especially problematic when data is shared between operators and organisations. Under the current paradigm, the sender must trust that the receiver correctly implements the necessary controls. They also accept the risk that the receiver may not understand the policies in the same way or may fail to implement them correctly.

A significant vulnerability is that once data leaves an operator's system, it loses all associated controls and embedded protections. Without persistent controls that travel with the data, the data becomes vulnerable to unauthorised access and misuse when transferred to another system or entity. This lack of persistent controls undermines the overall effectiveness of data protection measures.

Overall, these shortcomings highlight the challenges and limitations of the current paradigm of data sharing. The need for human interpretation and implementation of controls, reliance on individual operators, and loss of controls post-transfer create significant vulnerabilities and inconsistencies in data protection. These issues underscore the need for a more automated and integrated system to ensure consistent and effective data protection. This is summarised below (Table 1).

Domain	Limitations of current paradigm
Governance and policy	Broadly guided by the Privacy Act 1988 and the Australian Privacy Principles

	High level: no mandated specific technical standards for data sharing formats or interoperability. **Fragmented:** No comprehensive data sharing mandate that integrates with sector specific laws and no consistency across data sharing agreements/MOUs. **Siloed:** Compliance generally managed organisation-by- organisation. **Inconsistent**: Each body interprets regulatory requirements differently. **Lack of maturity**: Ethical standards is still evolving; fairness and transparency vary between data custodians.
Identity	**Fragmented**: Multiple service platforms, separate logins for different applications. **Limited interoperability**: State, federal and private sector systems have different authentication systems.
Data modelling	**Inconsistent**: Each organisation may handle data in its own schemas; no mandated common data models or ontologies; machine to machine data exchange is difficult, though some sectors do have their own data standards (e.g. Healthcare data standards). **Inefficient**: Manual interpretation of data field names, measurement definitions, and decoding of classification of systems; different data schemas and large variations in metadata quality; slow integration and analysis.
Integration and interoperability	**Siloed**: Agencies / organisations operate their own data silos; some progress in banking, but CDR is not universally adopted. **Unstable**: Heavy reliance on custom agreements and/or point-to-point negotiation; often one-off, time consuming, not future-proof when parties change their data formats or systems. **Inefficient**: Manual / semi-manual ETL (Extract Transform Load) processes; significant effort and duplication of work across organisation when similar data is shared.

Table 1. Key limitations of current data sharing paradigm.

3.3 Problem statement

The central problem is the reliance on users to navigate these risks independently. Most users are not equipped to manage these complexities, yet the burden falls on them. Because most users lack the specialised knowledge and skill to manage these complexities, the burden is untenable. We need a more automated and integrated system that embeds controls within both the data itself and the system architecture, ensuring persistent protection while reducing the burden on users.

This system should provide robust protections, ensuring consistent and effective data protection without relying on individual user expertise.

4 A New Paradigm

4.1 Vision: The system protects the user

This paper explores the potential evolution of the Trusted Data Sharing framework into a digital infrastructure that redefines data sharing and decision-making. Our proposed infrastructure addresses complex data sharing challenges across diverse ecosystems by minimising manual intervention and reliance on users' experience and knowledge.

Through automation and machine-interoperable policies, the system ultimately protects the user while streamlining operations.

The framework leverages metadata, sharing rules, and digital controls to systematically assess and mitigate data sharing risks. By semi-automating complex evaluation processes, this approach seeks to reduce the mental and administrative burden on data operators while enabling safer, more confident data sharing.

Data sharing capabilities are maturing globally. The European Health Data Space initiative is creating unified frameworks for secure, interoperable data exchange, while the Australian Research Data Commons (ARDC) is piloting similar concepts. These developments accelerate progress towards a unified, secure data sharing environment benefiting industry, government, operators, and citizens.

Our proposed infrastructure is designed to be:

- Adaptable to diverse data types and organisational contexts
- Clear and actionable without technical and legal complexity
- Capable of interpreting and implementing machine-readable policies with appropriate controls
- Powered by automated risk and harm assessment
- Enabling real-time, informed decision-making
- In an ‘always-on’ state

In this paper, we explore a system where metadata captures information throughout a data product’s complete lifecycle. From inception of a data product, metadata would be collected and stored to describe the data product’s quality, format, collection/creation conditions, access and use history, transformation history, and other relevant characteristics.

The metadata accumulates as the data product moves across its lifecycle and across different environments with varying levels of control, informing systematic risk assessment.

Through this continuous metadata collection, automated decision processes can analyse the information at each stage to mitigate data sharing risks, reducing the responsibility placed on individual data operators.

Our proposed infrastructure aims to do this in a generalised way that can be extended beyond the scope of our use case. The new paradigm is described in the table below (Table 2).

Domain	Description of new paradigm
Governance and policy	**Unified**: A cross-industry consortium (government, industry and user advocacy) establishes a unified set of rules, best practices and data ethics.

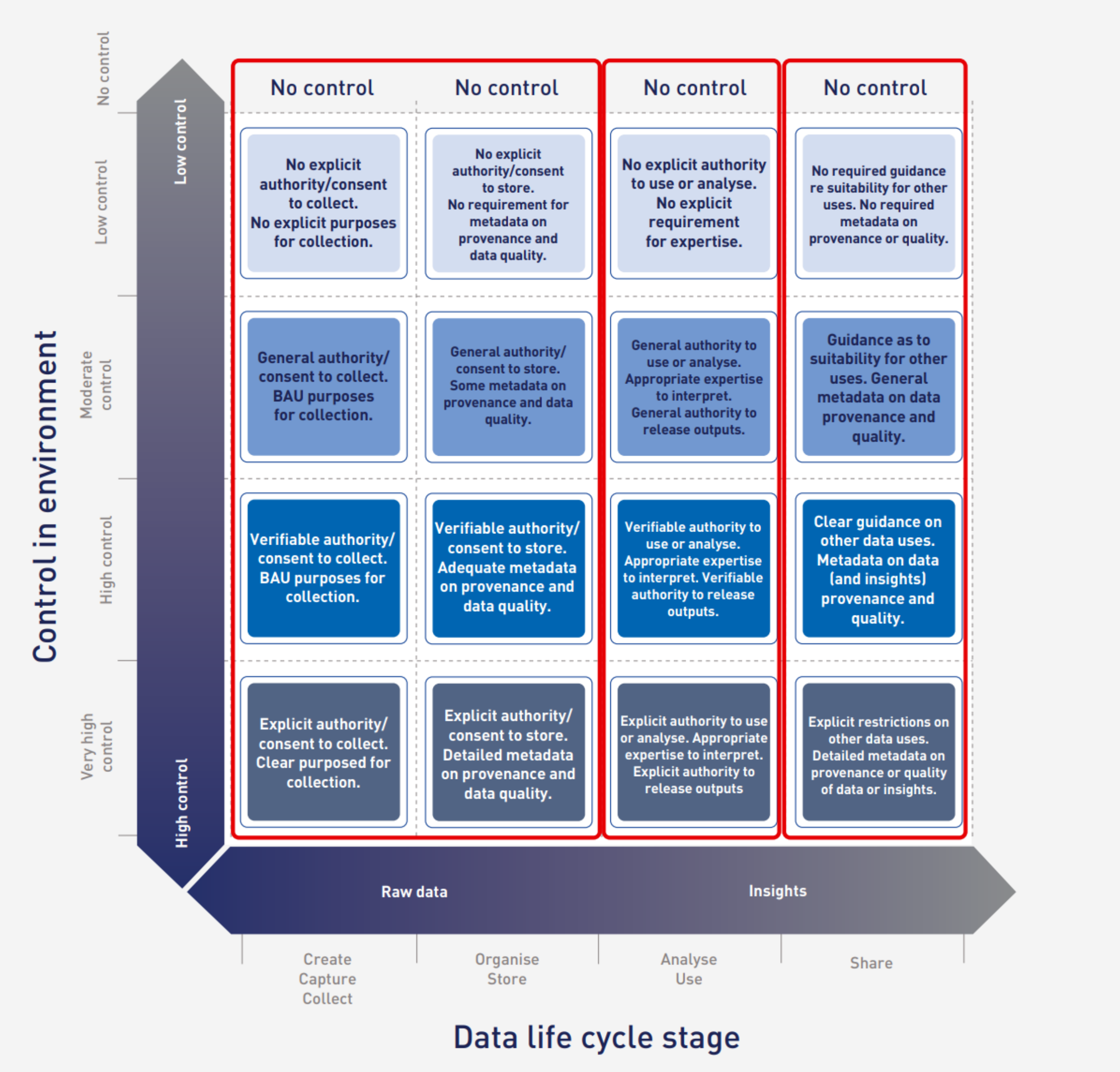
	Machine readable polices: Policies and rules and standards codified in machine readable formats, enabling automated enforcement and reducing ambiguity and inefficiencies that can also be included in schedules to data sharing agreements. **Continuous improvement**: Alignment of regulation with evolving technology and user needs; updated policies can be easily enforced through machine-interpretable formats.
Identity	**Integrated**: Users authenticate once through a trusted provider to access data from multiple systems. **Secure:** Trusted technologies to ensure robust authentication of identities. **Dynamic:** Identity systems dynamically update when user circumstances change, improving security and improving user experience.
Data modelling	**Consistent:** Common data models and ontology defined by industry standards, ensuring consistent interpretation across providers and end users. **Efficient:** Machine-readable metadata enables AI to support data classification, validation, and integration, reducing manual effort. **High quality:** Automated checking of incoming data against established formats and constraints enforces continuous improvement of data quality for trusted data sharing.
Integration and interoperability	**Timely**: When desired, providers can expose data through uniformly defined APIs that adhere to open standards, ensuring authorised users/organisational recipients can access data in a timely fashion. **Integrated:** Integrated data enables sophisticated queries to be run (e.g., integrated view of information/data patterns across multiple organisations to provide a holistic view). **Interoperable**: Simple process to add new providers, adapt to regulatory requirements, scale easily, and integrate new data sources.

Table 2. Key features of the new paradigm.

4.2 The data product is at the centre of the new paradigm

Firstly, we need to establish what constitutes a data product in this context.

A data product is any piece of digital information that moves through the data lifecycle as outlined in the Frameworks and Controls for Data Sharing paper – from creation and storage through analysis, use, and sharing with other parties.



	Create Capture Collect	Organise Store	Analyse Use	Share
No control	No control	No control	No control	No control
Low control	No explicit authority/consent to collect. No explicit purposes for collection.	No explicit authority/consent to store. No requirement for metadata on provenance and data quality.	No explicit authority to use or analyse. No explicit requirement for expertise.	No required guidance re suitability for other uses. No required metadata on provenance or quality.
Moderate control	General authority/ consent to collect. BAU purposes for collection.	General authority/ consent to store. Some metadata on provenance and data quality.	General authority to use or analyse. Appropriate expertise to interpret. General authority to release outputs.	Guidance as to suitability for other uses. General metadata on data provenance and quality.
High control	Verifiable authority/ consent to collect. BAU purposes for collection.	Verifiable authority/ consent to store. Adequate metadata on provenance and data quality.	Verifiable authority to use or analyse. Appropriate expertise to interpret. Verifiable authority to release outputs.	Clear guidance on other data uses. Metadata on data (and insights) provenance and quality.
Very high control	Explicit authority/ consent to collect. Clear purposed for collection.	Explicit authority/ consent to store. Detailed metadata on provenance and data quality.	Explicit authority to use or analyse. Appropriate expertise to interpret. Explicit authority to release outputs	Explicit restrictions on other data uses. Detailed metadata on provenance or quality of data or insights.

Figure 1. Matrix illustrating varying levels of control across different stages of the data lifecycle.

The data product could take the form any known data type or format, for example:

- **Tabular Data**: Organised in rows and columns, like spreadsheets or database tables.
- **Numeric Data**: Quantitative data, such as integers, decimals, and percentages.
- **Text Data**: Written content, including articles, emails, PDF documents, Word documents, and social media posts.
- **Images**: Visual data, such as photographs, diagrams, and scanned documents.
- **Audio**: Sound recordings, including music, podcasts, and voice memos.
- **Video**: Moving images, such as movies, video clips, and animations.
- **Geospatial Data**: Information related to geographic locations, like maps and GPS coordinates.

- **Sensor Data**: Data collected from sensors, such as temperature readings, humidity levels, and motion detection.

As a data product moves through its lifecycle, it traverses environments with varying levels of control, as detailed in the Frameworks and Controls for Data Sharing paper.

A fundamental concept is that when a data product that is shared and/or transformed, it **creates a new data product derived from the original**.

Consider email creation and transmission as an example:

1. *A user initiates a 'New email' in Outlook*
2. *A blank email is created, potentially with the user's signature and associated metadata - this represents the creation of the initial data product*
3. *The user composes the email and adds recipients, modifying the original data product (the blank email).*
4. *When the user sends the email, multiple new data products are created:*
    - *A copy stored in the sender's account (locally, in the cloud, or both)*
    - *Separate copies delivered to each recipient's account*

For the new paradigm to function effectively, the system must track the complete history of data product creation, transformation, and sharing. Each new instance of the data product (like the recipients' copies of the email) must inherit the original's history and maintain records of its relationship to the source, including any subsequent transformations that differentiate it from the original.

4.3 How this could be achieved

Metadata – data about data – describes fundamental properties of data including:

- Quality metrics and format specifications
- Collection methods and timing
- Contextual information
- Data accuracy, completeness, and consistency
- Environmental conditions during collection
- Encoding specifications

Beyond these basic properties, metadata also captures the data's journey:

- Applicable rules and regulations during collection
- Access and custody history
- Usage purposes
- Transformation records

As illustrated in Figure 2, when new data products are created and shared, their metadata evolves to document the complete lifecycle of each data product.

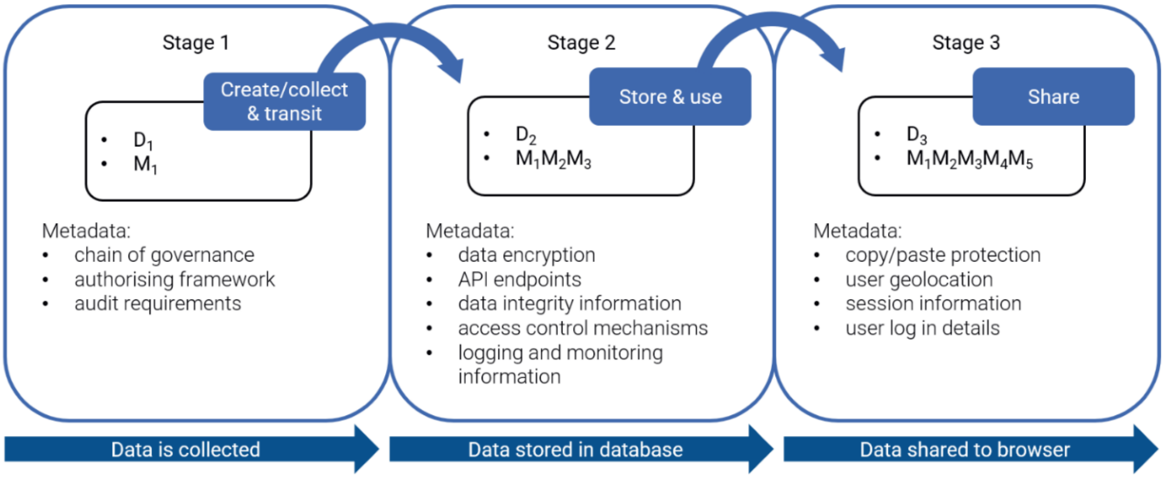


Figure 2. Diagram illustrating the evolution of metadata as it moves through the data lifecycle.

As data products progress through their lifecycle – from creation and collection through storage, use, sharing, and archival – they accumulate metadata that documents their journey.

The metadata captures critical information including:

- Data quality metrics
- Collection conditions
- Format specifications
- Required protection measures (e.g., disabling screenshot functionality or to toggle collapsing cells to their collapsed state)

During each lifecycle stage, data may undergo changes that could affect its integrity, such as data loss, compression artifacts, or corruption. This comprehensive metadata enables both users and software to:

- Track the data product's journey and transformations
- Identify derivative data products
- Determine sharing permissions
- Define appropriate use cases

As a result, digital controls can interpret this metadata to make decisions about the data product’s future use, such as:

- Make automated decisions about data sharing
- Provide risk assessment guidance to users who wish to share the data further
- Monitor escalating risks for specific users or entities

Implementation requirements

To achieve this in practice, new or adapted software must:

- Facilitate metadata accumulation
- Enable metadata interpretation, and
- Execute actions based on metadata under policy guidance.

Possible implementation options include:

1. Developing new software and standards for consistent tracking of data product characteristics
2. Adapting existing metadata managements solutions to perform the above.

Example: Screenshot Prevention of Sensitive financial information Capture

Consider a security risk where users taking screenshots of sensitive financial information on their smartphones could become vulnerable to scams.

Current paradigm:

Each financial institution must individually implement screenshot prevention by modifying their webpage parameters. This fragmented approach relies on individual institutions to:

- Recognise the security risk
- Implement appropriate controls
- Maintain consistent protection across their digital assets

New paradigm:

The system automatically protects sensitive financial information through metadata-driven controls:

1. The financial information contains metadata identifying it as requiring screenshot protection
2. The operating system or browser reads this metadata
3. Screenshot prevention controls activate automatically when this content is displayed.

This approach shifts from individual, hardcoded solutions to a systematic, metadata-driven protection model. The system adapts to the data product's requirements, rather than requiring each institution to implement bespoke controls. Most importantly, it automatically protects users without relying on individual

4.4 What technology is needed and how it could work

The new data sharing paradigm requires four key components:

1. **Unified governance framework:** A cross-industry body that establishes and maintains unified rules, best practices, and ethical data standards to ensure consistent policy application across the ecosystem and build trust in data sharing.
2. **Metadata management and storage:** Systems that track and preserve the complete history of data products, including transformations and essential attributes throughout their lifecycle.
3. **Machine-readable policies**: Policies codified in formats like JSON that enable direct system action based on clearly defined rules, eliminating ambiguity in interpretation.
4. **Automated real-time decisioning**: Systems that can act immediately on metadata-driven policies, implement protective controls and enforce security measures automatically.

These components, which are all implementable with existing technologies and have been adopted in specific use cases, work together to create an adaptive system that protects users while maintaining data utility.

4.4.1 Data product architecture

The architecture of this system centres around data products, as illustrated in Figure 3. A data product combines two key elements:

- Structured data values
- Associated metadata including terms, conditions, update logs, and digital signatures.

Each data product's structure and metadata schema are purposefully designed for specific users and use cases, recognising that its value varies based on user context and needs. While metadata includes usage rules, these may be presented separately to users for clarity despite being integral to the metadata structure.

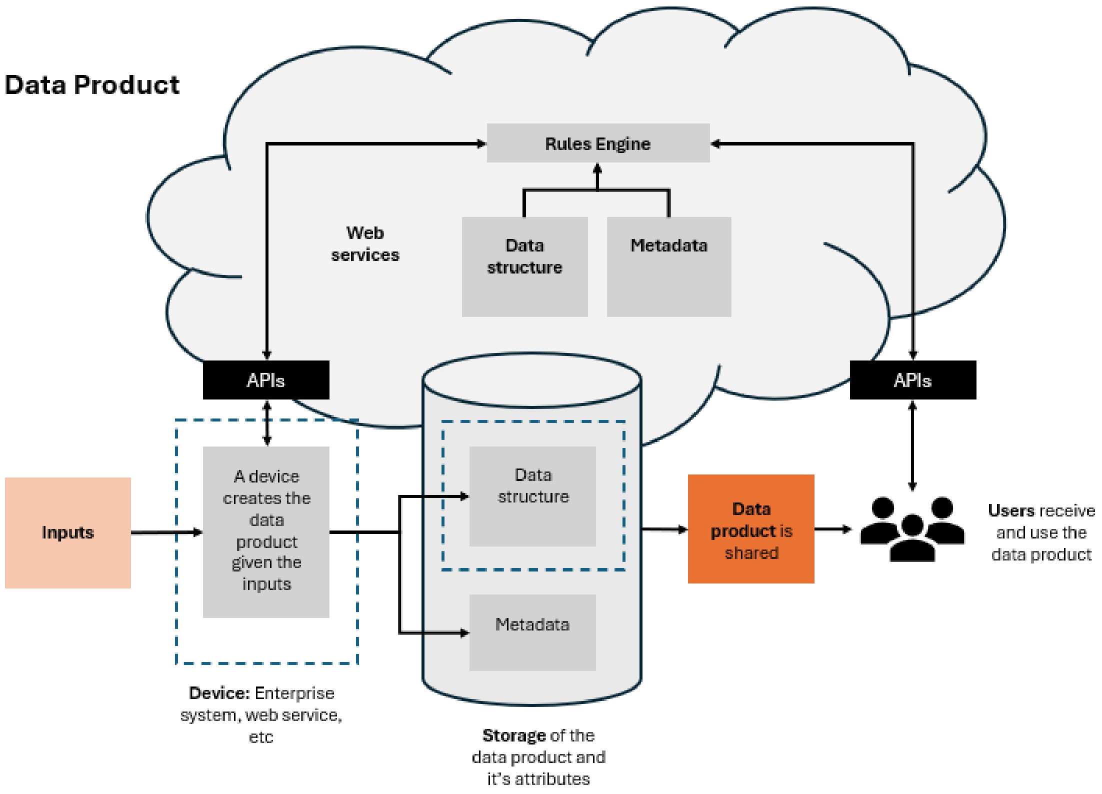

Data Product
Rules Engine
Web services
Data structure
Metadata
APIs
APIs
Inputs
A device creates the data product given the inputs
Data structure
Metadata
Data product is shared
Users receive and use the data product
Device: Enterprise system, web service, etc
Storage of the data product and it's attributes

Figure 3. Diagrammatic example of a cloud-based data product architecture.

Processing and storage flexibility

Processing and storage of data products can occur on various platforms, such as devices, computers, or cloud assets. This flexibility ensures the system can adapt to different organisational needs and infrastructure requirements.

In the case of a cloud-based architecture, a data product model would comprise three key architectural components:

1. Input processing

Inputs comprise any information that constitutes the data product - for example, an email's recipients, subject, body text, signature, and security tags. These inputs are initially processed through a device or system that creates the data product.

2. Cloud infrastructure

The cloud environment houses several critical components:

- Data and metadata storage
- Rules Engine configured by the product designer to enforce owner policies
- Web services for automated logging of data changes, digital signing, and updates
- APIs for secure data access and updates

3. User access

End users interact with the data product through APIs that enforce access rules based on:

- User permissions
- Data sensitivity
- Usage context
- Extraction requirements (e.g. whole records or extracts, depending on rules)

This architecture supports adaptable implementation while maintaining security and control through metadata-driven policies. The Rules Engine ensures that all data access and modifications align with established governance frameworks, creating a secure and auditable environment for data sharing.

5 A worked example: Unified academic credential sharing for job applications

ISO/IEC 5207 provides a template to describe a use case and all its considerations. In this example we will apply the data to the standard.

5.1 Use Case Description

In the current paradigm, the process of sharing educational credentials is fragmented and inconsistent. Each educational institution follows its own rules and regulations, leading to a lack of cohesion and trust in the authenticity of shared credentials. A typical workflow for a student applying for a job is outlined below:

1. *Log into their university's portal*
2. *Navigate to the section for academic credentials*
3. *(Often) pay a fee to collect a PDF of the transcript*
4. *Upload the PDF to the job application form.*

Credentials and academic data are published in various formats, making integration and understanding challenging. Manual interpretation of data fields and schemas slows down the process, and students face a high cognitive burden gathering and managing their credentials.

Furthermore, employers are forced to trust the uploaded PDF. There are no built-in systems to verify that a student has not tampered with the PDF prior to uploading. This fragmented approach makes it difficult to verify credential authenticity without contacting each institution separately.

In the new paradigm, digital academic credentials become machine-readable and verifiable through a unified governance framework established by a cross-industry consortium. This ensures consistent policy application and enhances trust in shared credentials.

A potential workflow is outlined below:

1. *Authenticate through a trusted provider to a central portal*
2. *Follow a simple, navigable UI (with tips from an AI assistant) to the credentials section*
3. *Either: a. Download an offline transcript secured by digital signature, and then b. Upload to job application site*
4. *Or: a. Input job application details in a form to have credentials delivered directly to the employer.*

Students authenticate once through a trusted provider, gaining secure, seamless access to their credentials. Common data models and machine-readable metadata enable efficient integration and understanding of credentials from different sources. These interoperable systems facilitate timely credential sharing while providing a unified user experience that reduces cognitive burden on students.

Intelligent AI assistance offers explanations and support, simplifying credential management and sharing with minimal manual effort. This new approach builds greater trust in the authenticity and reliability of educational credentials, benefiting both students and employers.

5.2 Application of the ISO/IEC 5207 template

5.2.1 Use case name and overview

Unified Academic Credential Sharing for Job Applications

This use case describes a system where students can securely access and share their academic credentials with employers through a unified, trusted platform.

5.2.2 Domain areas

This use case pertains to the domains of education, employment, and digital identity management.

5.2.3 Objectives

The primary objective is to streamline academic credential sharing. Key benefits include enhanced trust in credential authenticity, reducing administrative burden on students, and improved employer verification processes. The scope applies to all participating educational institutions and employers within the unified credential sharing system but is limited to academic credentials and does not cover other types of personal data.

5.2.4 Narrative

Students access a central portal through single authentication with a trusted provider. From there, they can either download transcripts secured with trust signatures or authorise direct delivery to employers. Common data models and machine-readable metadata ensure consistency and integration across institutions. AI assistance guides students through the process, reducing cognitive burden while maintaining data integrity.

5.2.5 Data lifecycle stages

The data lifecycle stages include:

- The collection of students' academic data by educational institutions
- Secure storage in a centralised database
- Access by secure authentication
- Sharing of credentials with employers through secure APIs, and
- Dynamic updates to the system whenever there are changes in student data, such as a name change.

5.2.6 Figures

Figures related to this use case include a data flow diagram illustrating the flow of data from educational institutions to the central portal and then to employers, and a sequence diagram showing the steps involved in student authentication, data access, and sharing.

5.2.7 Stakeholders and stakeholder considerations

The key stakeholders and key considerations are outlined below (Table 3):

Stakeholder	Key considerations
Students	Need a user-friendly, secure way to share credentials
Educational institutions	Must standardise data formats and ensure data integrity
Employers	Require reliable and verifiable credentials
Government agencies	Oversee compliance and integration with public services
Technology providers	Develop and maintain the technical infrastructure

Table 3: Key stakeholders and considerations for a unified academic credential sharing use case.

5.2.8 Data characteristics

Key data characteristics include structured data such as academic records and transcripts, and metadata that includes data provenance, digital signatures, and update logs.

5.2.9 Key performance indicators

Key performance indicators for this use case include:

- User satisfaction measured through student and employer feedback
- Data integrity indicated by the number of verified credentials without discrepancies
- System uptime reflecting the availability of the central portal.

5.2.10 Challenges and issues

Challenges and issues include:

- Ensuring all institutions adhere to common data models
- Protecting student data from unauthorised access, and
- Integrating with various institutional systems.

5.2.11 Societal concerns

Ensuring digital inclusion remains critical so all students can access the system, regardless of technological resources or abilities. Additionally, maintaining confidentiality and security of student data is paramount to prevent misuse or unauthorised access.

5.2.12 Data security, privacy, and trustworthiness

Data security, privacy, and trustworthiness are ensured through multi-factor authentication, encryption of data during storage and transfer, and the use of digital signatures to ensure the integrity and authenticity of credentials.

5.2.13 Key insights

Key insights include the efficiency of the streamlined process in enhancing trust in credential authenticity, and the system's scalability to incorporate additional institutions while adapting to regulatory changes.

5.3 The use case under the current paradigm

In the current paradigm, the sharing of academic credentials—such as transcripts or certifications—remains primarily manual and fragmented. Students, institutions, and employers navigate a disjointed landscape lacking integration and automation, resulting in inefficiencies, delays, and increased risks.

5.3.1 Process Description

Figure 4 illustrates the typical process of academic credential sharing under the current paradigm involves several disconnected steps.

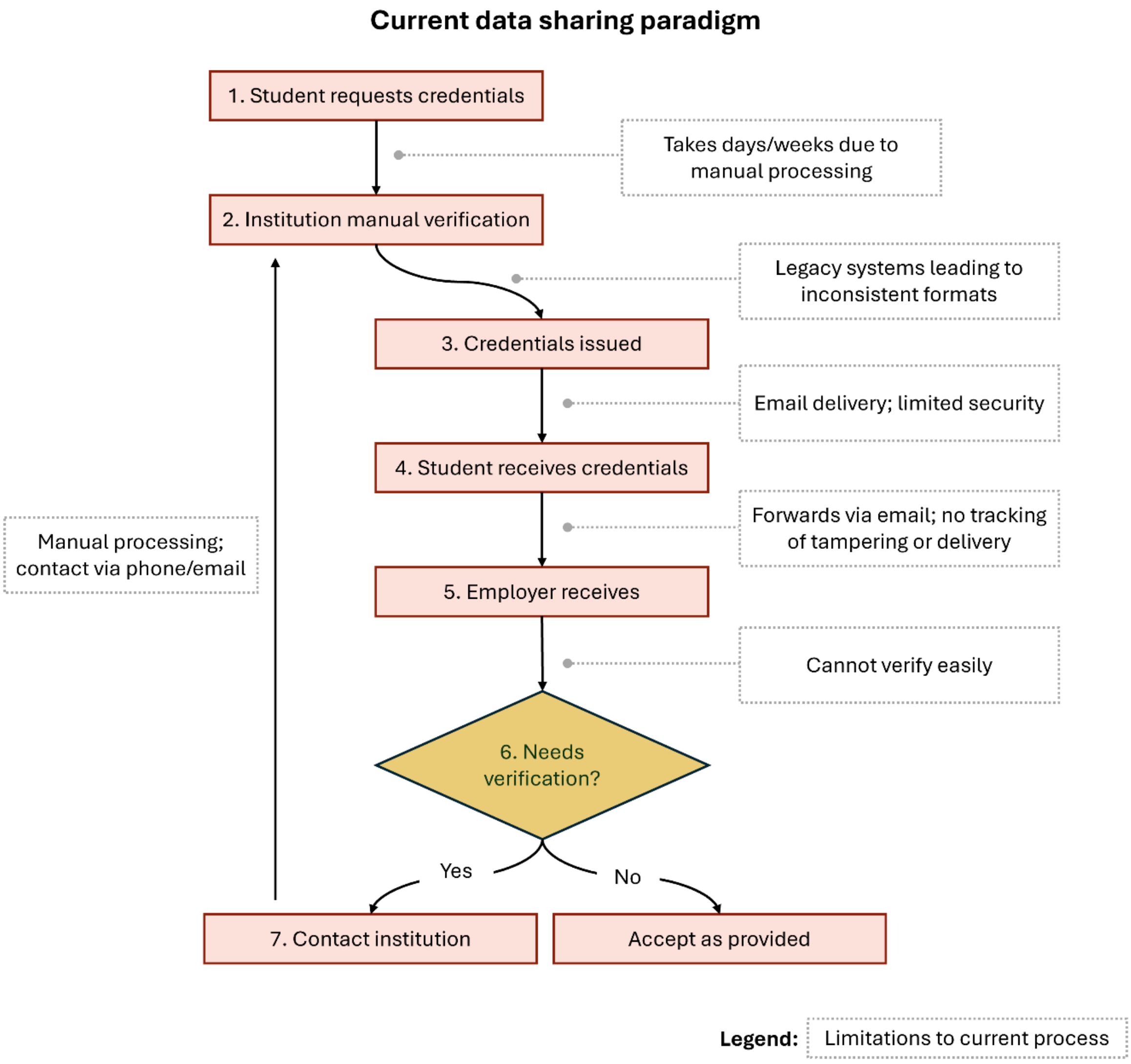

Current data sharing paradigm
1. Student requests credentials
Takes days/weeks due to manual processing
2. Institution manual verification
Legacy systems leading to inconsistent formats
3. Credentials issued
Email delivery; limited security
4. Student receives credentials
Forwards via email; no tracking of tampering or delivery
Manual processing; contact via phone/email
5. Employer receives
Cannot verify easily
6. Needs verification?
Yes
No
7. Contact institution
Accept as provided
Legend:
Limitations to current process

Figure 4. Schematic workflow illustrating the process of academic credential-sharing under the current data sharing paradigm.

5.3.2 Limitations and Inefficiencies

The current paradigm of data sharing in academic credentialing systems is fraught with limitations and inefficiencies that hinder effective data management and protection. These systems are often siloed across institutions, leading to a lack of integration and standardisation. As a result, each institution operates independently, storing credentials in various formats and requiring manual intervention to standardise documents for sharing.

This fragmentation not only creates inefficiencies but also increases the operational overhead, as administrative resources are needed to manage requests, verify credentials, and process physical documents. The reliance on manual processes further exacerbates these issues, leading to delays, errors, and increased costs.

Moreover, the security risks associated with current data sharing methods cannot be overlooked. Sharing credentials via email or other unsecured methods introduces significant vulnerabilities, as documents can be intercepted, tampered with, or accessed by unauthorised parties.

The lack of encryption and secure transmission protocols makes these methods susceptible to breaches, while inconsistent sharing practices further compound these risks. Additionally, the absence of clear audit trails complicates efforts to ensure accountability, making it difficult for institutions to track who accessed a credential and when. These limitations highlight the need for a more integrated, automated, and secure data sharing framework.

The following table summarises key challenges by domain (Table 4):

Domain	Key challenges under the current paradigm
Governance and policy	**Fragmented Compliance**: Institutions operate independently, leading to inconsistent interpretations of regulatory requirements and fragmented compliance efforts. **Manual Processes:** Reliance on manual intervention for credential verification creates bottlenecks, delays, and increased operational costs. **Limited Accountability**: Lack of clear audit trails complicates efforts to track credential access and ensure accountability.
Identity	**Siloed Systems**: Multiple service platforms and separate logins create a fragmented identity landscape, lacking a universally accepted digital identity. **Inconsistent Security**: Sharing credentials via unsecured methods like email introduces significant security risks, including interception and tampering.
Data Modelling	**Lack of Standardisation:** Institutions store credentials in various formats, requiring manual standardisation for sharing, leading to inefficiencies. **Data Quality Issues**: Manual data handling increases the likelihood of errors, miscommunications, and the need for reprocessing.
Integration and Interoperability	**Inefficient Verification**: Verification by third parties is slow and relies on manual checks, causing delays and lost opportunities. **Inconsistent User Experience**: Students face inconsistencies in the credential request and sharing process, leading to confusion and frustration. **Limited Transparency**: Institutions lack tools to effectively track credential usage, making it difficult to ensure accuracy and legitimacy.

Table 4: Summary of key challenges faced by academic credentialling systems under the current paradigm.

5.4 Use case under the new paradigm

In the new paradigm, academic credential sharing is transformed into a highly automated, standardised, and secure process. This approach leverages machine-readable policies, metadata-rich frameworks, and advanced governance standards as outlined in ISO/IEC 5212. These innovations eliminate inefficiencies, ensure data integrity, and empower users by providing greater control and transparency.

5.4.1 Process Description

The reimagined process, outlined in Figure 5, integrates advanced technologies and frameworks to streamline credential sharing while embedding protections for both data and users.

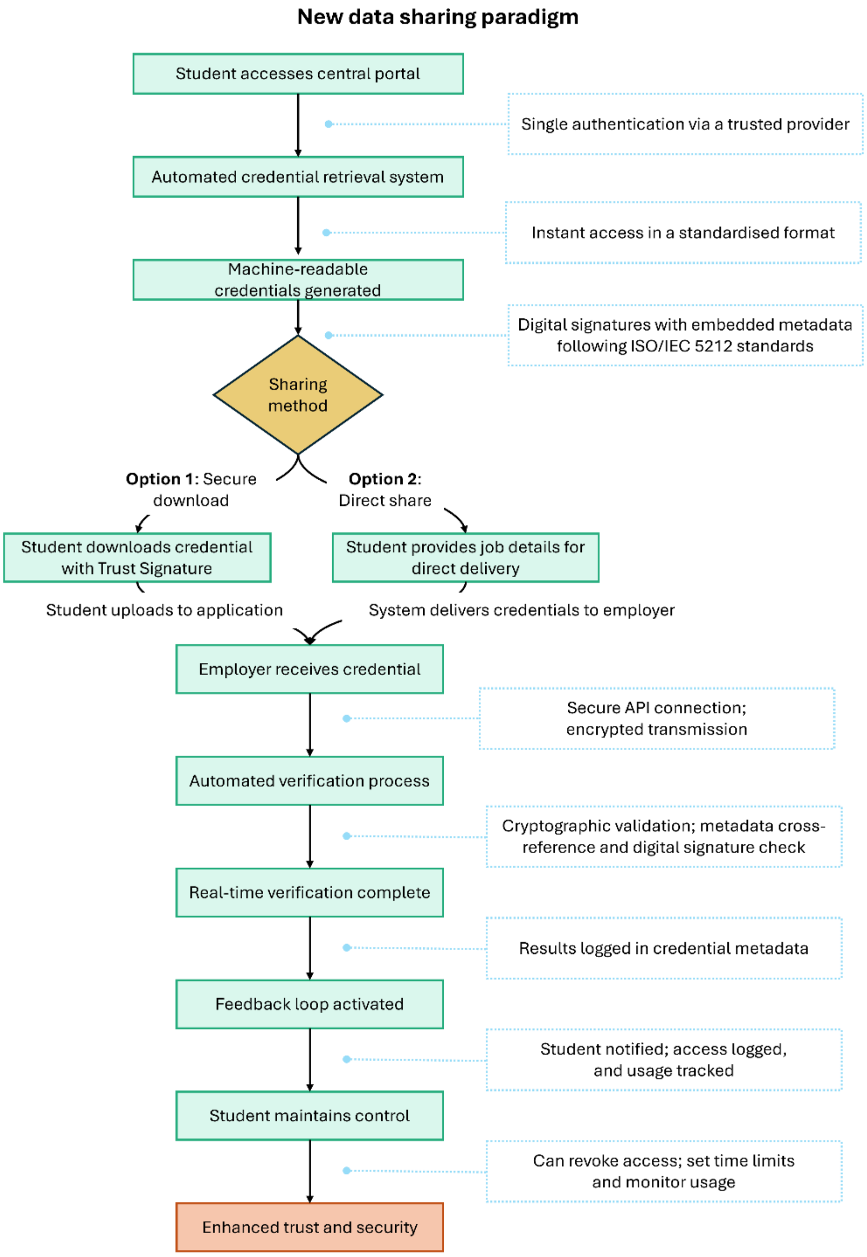

New data sharing paradigm
Student accesses central portal
Single authentication via a trusted provider
Automated credential retrieval system
Instant access in a standardised format
Machine-readable credentials generated
Digital signatures with embedded metadata following ISO/IEC 5212 standards
Sharing method
Option 1: Secure download
Option 2: Direct share
Student downloads credential with Trust Signature
Student provides job details for direct delivery
Student uploads to application
System delivers credentials to employer
Employer receives credential
Secure API connection; encrypted transmission
Automated verification process
Cryptographic validation; metadata cross-reference and digital signature check
Real-time verification complete
Results logged in credential metadata
Feedback loop activated
Student notified; access logged, and usage tracked
Student maintains control
Can revoke access; set time limits and monitor usage
Enhanced trust and security

Figure 5. Reimagined academic credential-sharing process under proposed new data sharing framework.

5.4.2 Benefits of the new paradigm

The transition from the current fragmented approach to a unified, metadata-driven framework delivers substantial improvements across all aspects of credential sharing. Table 5 summarises key paradigm benefits by domain.

Domain	New paradigm benefits

Governance and policy	**Unified governance:** A cross-industry consortium, including government, educational institutions, and user advocacy groups, establishes a unified set of rules and best practices for credential data sharing.
	Consistent policies: Policies are consistently applied across all educational institutions, ensuring uniform compliance and reducing ambiguity.
	Machine-readable policies: Policies and standards are codified in machine-readable formats, allowing for automated enforcement and reducing inefficiencies. For example:
	`Policy {`
	`  ID: "edu-credential-sharing"`
	`  Rules {`
	`    if (provider in ["provider", "OAuth2"]) { auth: required }`
	`    if (format in ["JSON", "XML"]) { format: enforced }`
	`    if (signature == "digital") { integrity: required }`
	`  }`
	`}`
	Continuous improvement: Regulations are regularly updated to align with evolving technology and user needs, with changes easily enforced through machine-learnable formats.
	Enhanced trust: A unified approach builds greater trust in the authenticity and reliability of shared educational credentials.

Identity	**Integrated identity systems**: Students authenticate once through a trusted provider to access their credentials from multiple educational institutions. The data is encrypted during transfer, and students manage their credentials through a central portal with AI assistance. **Secure authentication**: Advanced technologies ensure robust and secure authentication of student identities **Dynamic updates**: Identity systems automatically update when student circumstances change, improving security and user experience. Students authenticate securely using multi-factor authentication (MFA). If a student legally changes their name, the system automatically updates their profile across all connected educational institutions, ensuring credentials are always current and accurate.

Data modelling	**Consistent data models**: Common data models and ontologies are defined by industry standards, ensuring consistent interpretation of credentials across institutions and employers. For example:
	`Ontology {`
	`  Student {`
	`    Name: String,`
	`    StudentID: String,`
	`    DateOfBirth: Date`
	`  }`
	`  Credential {`
	`    CredentialID: String,`
	`    Type: String,`
	`    IssueDate: Date,`
	`    Institution: String`
	`  }`
	`  Relationship {`
	`    EarnedBy: Credential -> Student`

	}
	}
	Machine-readable metadata: Metadata is machine-readable, enabling AI to support data classification, validation, and integration. For example:
	Metadata {
	CredentialID: "12345"
	StudentID: "67890"
	IssuedDate: "2025-01-20"
	Institution: "University of Sydney"
	CredentialType: "Bachelor of Science"
	Status: "Active"
	DigitalSignature: "abc123def456"
	
	Updates {
	Update1 {
	Date: "2025-05-15"
	Change: "Name Change"
	NewValue: "John Doe"
	}
	Update2 {
	Date: "2026-06-10"
	Change: "Graduation"
	NewValue: "Completed"
	}
	}
	}
	High-quality data: Automated checking of incoming credential data against established formats and constraints ensures continuous improvement of data quality.
	Format Validation: When a credential is received, the system automatically checks if the date is in the correct format (e.g., YYYY-MM-DD). If the date is not in this format, the system flags it for correction.
	Efficient integration: Consistent data models and high-quality metadata enable seamless and efficient integration of educational credentials from different sources.

Integration and interoperability	**Timely access:** Educational institutions can expose credential data through uniformly defined APIs that follow open standards, ensuring timely access to data. Educational institutions provide APIs for credential data access, following open standards. Below is an example workflow: - **API Endpoint**: e.g. https://api.university.edu/credentials - **Authentication**: Students use a secure website to get an access token. - **API Request:** GET /credentials?studentID=12345 with the access token. - **Data Retrieval:** API fetches data from the institution's database. - **Response**: Returns data in JSON format. - **Integration**: Employer's system verifies and integrates the data. **Integrated data**: Integrated data enables sophisticated queries and a holistic view of student credentials across multiple institutions. **Interoperable systems:** Simple processes to add new institutions and adapt to regulatory requirements, allowing for easy scaling and integration of new data sources. Below are two example workflows: 1. Adding a new institution: - **Onboarding request**: The new institution submits an onboarding request through the central portal.

	- **API integration**: The institution's IT team integrates their systems with the central API, following the provided documentation. - **Data standardisation:** The institution formats their credential data according to the common data model. - **Testing**: Conduct end-to-end testing to ensure data flows correctly and securely. - **Approval**: Once testing is successful, the institution is approved and added to the network. 2. Adapting to Regulatory Requirement Changes: - **Regulatory update notification**: The system receives a notification about a regulatory change. - **Impact analysis:** Assess how the change affects current processes and data models. - **Policy update:** Update the machine-readable policies to reflect the new requirements. - **System update**: Implement necessary changes in the system to comply with the new regulations. - **Testing and deployment:** Test the updates and deploy them across all connected institutions. - **Efficient sharing:** Integrated and interoperable systems enable timely and efficient sharing of educational credentials.

Table 5. Summary of key benefits of the new proposed data sharing paradigm by domain for academic credentialing systems.

The following table (Table 6) compares the current paradigm with the new approach, illustrating how automated, machine-readable policies can transform credential management and enable seamless and interoperable operations across the entire ecosystem.

Aspect	Current paradigm	New paradigm	Improvement
Credential request Process	Manual request through institution portals or by contacting staff.	Automated requests governed by machine-readable policies, enabling instant access.	Faster, policy-driven automation ensures seamless, accurate, and timely retrieval of credentials.
Data format and metadata	Fragmented formats; minimal metadata for interoperability.	Standardised, metadata-rich formats (e.g., JSON, XML) with lifecycle tracking.	Enhanced cross-platform compatibility, traceability, and consistent data representation.
Credential verification	Manual verification through emails, phone calls, or faxes.	Automated API-based verification using metadata and blockchain.	Instant, tamper-proof verification eliminates delays and ensures accuracy.
Data integrity and Security	Reliance on paper or email sharing, vulnerable to tampering or interception.	End-to-end encryption, digital signatures, and blockchain for tamper-proof credentials.	Credentials are tamper-proof and secure, with robust encryption and verifiable audit trails.
Policy enforcement	Ad hoc human enforcement of sharing and access rules.	Embedded machine-readable policies automate access and sharing restrictions.	Policies are consistently enforced without human intervention, reducing

			errors and ensuring compliance.
User protection	Limited user control: credentials can be shared multiple times without oversight.	Students control access, receive real-time notifications, and can revoke permissions instantly.	Enhanced privacy and user control reduce unauthorised sharing risks and empower students.
Credential sharing	Shared via email or paper, risking delays, misdelivery, or tampering.	Shared via secure, metadata-aware APIs or encrypted links with time-limited access.	Secure, policy-governed sharing ensures only authorised access and eliminates risks of tampering.
Audit and tracking	Limited tracking of credential sharing or usage; manual audits.	Comprehensive metadata audit trails track every action in real-time.	Enhanced transparency and accountability with detailed logs of access, sharing, and lifecycle events.
Interoperability	Fragmented systems; no standardisation for cross-institutional sharing.	Standardised APIs and metadata frameworks enable seamless interoperability.	Institutions, students, and third parties benefit from globally compatible credential sharing.
Error rate	High potential for human error in manual data handling.	Automation and metadata-driven processes eliminate manual errors.	Significant reduction in errors, ensuring accurate credential preparation and verification.
User experience	Inconsistent interfaces and procedures across institutions.	Intuitive platforms with one-click sharing, real-time tracking, and revocation controls.	Simplified, user-friendly experience for students and recipients alike.
Cost efficiency	High administrative overhead due to manual processes.	Lower costs with automated workflows and metadata-driven governance.	Institutions save time and resources by minimising manual intervention and streamlining operations.
Scalability	Delays during peak periods (e.g., graduations).	Scalable system handles high volumes of requests without performance degradation.	Reliable, efficient operations even during high-demand periods.
Global compatibility	Difficult to share credentials across jurisdictions due to varying standards.	Metadata standardisation enables seamless cross-border credential sharing.	Students can apply for global opportunities effortlessly, with credentials easily interpreted worldwide.

Table 6. Table summarising key comparative differences between the current and proposed data sharing framework in the context of academic credential-sharing.

5.5 Risks and Harms of Sharing Data

Data sharing presents multifaceted risks with significant real-world consequences. These harms stem from various aspects of data product applications—from unauthorised access to personal information to unintended identification and trust erosion. The severity ranges from reversible incidents to permanent damage, as exemplified by inadequate security exposing individuals to identity theft with lasting consequences.

All data product applications require vigilant monitoring for potential harms. This includes continuous output evaluation and result testing to prevent unintended consequences. Quantifying secondary harms, unanticipated benefits, and long-term community impacts is essential. Testing itself may cause harm, particularly when systems make consequential decisions—a risk that must be addressed even with consenting human testers.

Moreover, changing the context or environment of data product usage can trigger unforeseen consequences. Therefore, usage modifications require careful consideration and rigorous monitoring to mitigate potential harms. By comprehensively understanding these risks, we can navigate data sharing complexities while ensuring data-driven systems deliver benefits without compromising ethical standards or harming individuals and communities.

The table below (Table 7) shows a non-exhaustive list of potential harms associated with data sharing and listed in the NSW AI Assessment Framework.

Potential Harm	Description
Physical harms	Physical harms to the user or other entities.
Psychological harms	Stress or distress caused to individuals due to errors or misuses of the system.
Environment harms or harms to the broader community	Environmental harms such less funding for environmental programs.
Unauthorised use of health or sensitive personal information	Improper access to or misuse of sensitive personal or data without consent.
Impact on right, privilege or entitlement	the burden of data protection is on users, leading to potential misuse and unauthorised access to sensitive information and discrimination.
Unintended identification of misidentification of an individual	Misidentification of individuals or their credentials due to system errors.
Financial or commercial impact	Financial harm caused by credentialing errors or operational inefficiencies.
Incorrect advice or guidance	Consequences associated with receiving falsified information and broken trust.
Inconvenience or delay	System inefficiencies causing delays in credential access or sharing.
Erosion of trust	Loss of trust in the system due to perceived or actual breaches of privacy.

Ethical implications	Use of the system in ways that raise concerns about fairness or bias.
Economic disruption / impact	Economic harm caused by breach of trust and slowdown of applicant review.
Social equality	The system disproportionately disadvantaging certain groups.

Table 7. Non-exhaustive list of potential harms associated with data sharing.

A new system must prioritise meaningful advancements in mitigating all potential harm risks to ensure ethical and effective deployment. The design should build trust through operational transparency and accountability while efficiently managing processes to prevent inconvenience or delays. Rigorous testing must identify and minimise as much as possible ethical concerns like bias or unfairness.

By providing accurate guidance, the system prevents economic impacts and social inequalities while incorporating mechanisms to monitor psychological, secondary, and cumulative harms. These features allow for prompt identification and mitigation of long-term adverse effects. Through these comprehensive protections, the system significantly reduces data sharing risks, fostering a safer, more reliable environment for users, organisations, and governments alike.

5.6 Risk Example: Fraud - Applicant falsifying academic credentials

Consider two branching paths:

1. A student is applying for the position truthfully and submits their academic transcript
2. A student is applying for the position but wants to gain the edge on the competition by falsifying their academic records before submitting their application

How can the employ trust the academic transcript they received is truthful?

5.6.1 Example Under the Current Paradigm

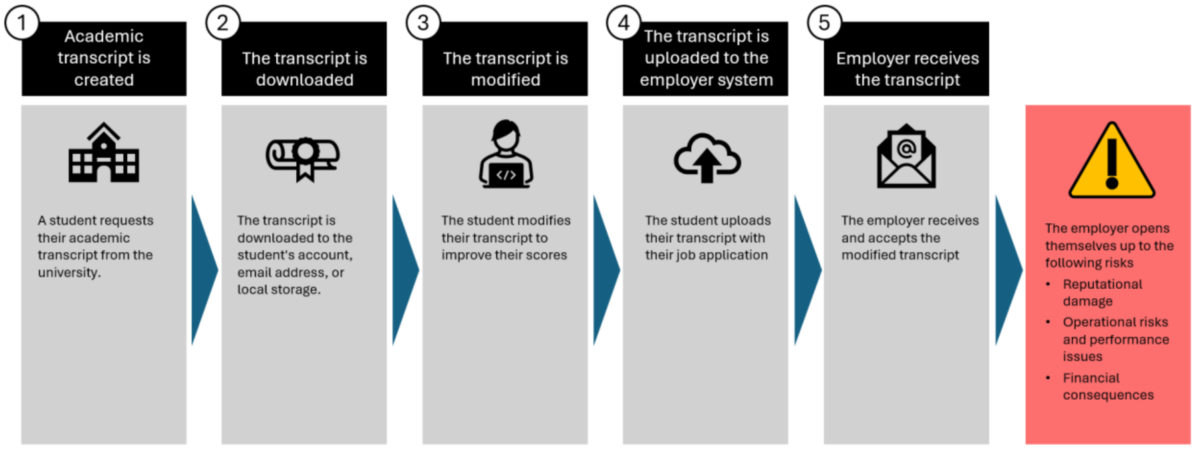
In the current paradigm, fraudulent credentials pose a significant risk due to the manual and fragmented nature of verification processes. The typical scenario unfolds when an applicant submits falsified paper transcript or digitally altered PDF to an employer (Figure 6). Without universal standardisation or security protocols, employers must independently verify authenticity by contacting issuing institutions.

While digital signatures can detect document alterations, they only benefit receivers who have established processes to evaluate signature validity and act accordingly. Institutions, lacking automation and real-time verification capabilities, rely on resource-intensive manual checks.

The verification dilemma creates a problematic choice: either accept credentials at face value due to time and resource constraints, or delay hiring decisions for weeks while awaiting proper confirmation—both options carrying significant risks, such as:

- **Operational risks:** Employers may hire unqualified candidates who cannot fulfill role requirements, resulting in financial losses and reputational damage.
- **Institutional risks**: Issuing institutions face credibility questions when their credentials prove easily forgeable.
- **Systemic risks:** Repeated fraud incidents undermine trust in academic systems, disadvantaging legitimate candidates who depend on verified records.

The limitation with current technologies is that they need to be put in place by each entity independently. It is the responsibility of the entity to ensure governance frameworks are followed and technologies are in place.



1
Academic transcript is created
A student requests their academic transcript from the university.
2
The transcript is downloaded
The transcript is downloaded to the student's account, email address, or local storage.
3
The transcript is modified
</>
The student modifies their transcript to improve their scores
4
The transcript is uploaded to the employer system
The student uploads their transcript with their job application
5
Employer receives the transcript
@
The employer receives and accepts the modified transcript
The employer opens themselves up to the following risks
• Reputational damage
• Operational risks and performance issues
• Financial consequences

Figure 6. Example scenario of a student tampering with an academic transcript and the risks it imposes to the employer under the current paradigm.

5.6.2 Example Under the New Paradigm

The new paradigm introduces a system activated by the data product itself rather than individual entities, ensuring controls and protections remain continuously active.

An applicant submits a digitally altered credential via the credential-sharing platform. The system automatically verifies the document by cross-referencing the metadata—which maintains a complete transformation history—with institutional records where the credential originated. Tampering is immediately detected, and the credential is flagged as invalid.

For example, if PDF elements representing academic results are modified, the transformation is captured in the metadata, similar to version control software. This becomes another documented step in the data product's lifecycle.

When the prospective employer receives the academic transcript, the system automatically analyses the document's lifecycle and identifies any modifications. Based on this analysis, the system can take several protective actions (Table 8):

Action	Description	Intervention example
Guide	Warn the user of potential risks associated with the received data.	Raise a warning with the employer that transcript can't be trusted, and a new one be requested from the applicant.
Restrict	Prevent recipients from using the data in ways that could be considered harmful.	The application is not allowed to be submitted by the candidate and therefore the employer never receives the falsified records. The applicant's email is automatically returned including the reason why and steps to resolve
Recommend	Provide options for safe use with suggestions for the most secure approach.	The application is recommended to be rejected and filtered out of the pool of legitimate candidates.

Table 8: Example interventions that could be actioned by a credential-sharing platform under the new paradigm.

In this situation, the system protects the company seeking to fill the new role and the education institution from potential reputational damage (Figure 7).

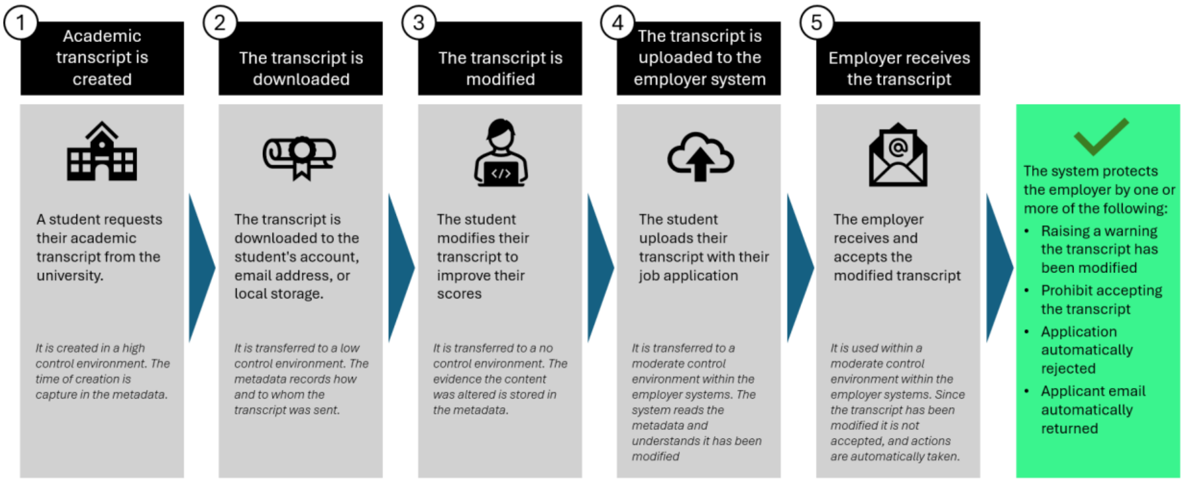

1
Academic transcript is created
A student requests their academic transcript from the university.
It is created in a high control environment. The time of creation is capture in the metadata.
2
The transcript is downloaded
The transcript is downloaded to the student's account, email address, or local storage.
It is transferred to a low control environment. The metadata records how and to whom the transcript was sent.
3
The transcript is modified
The student modifies their transcript to improve their scores
It is transferred to a no control environment. The evidence the content was altered is stored in the metadata.
4
The transcript is uploaded to the employer system
The student uploads their transcript with their job application
It is transferred to a moderate control environment within the employer systems. The system reads the metadata and understands it has been modified
5
Employer receives the transcript
The employer receives and accepts the modified transcript
It is used within a moderate control environment within the employer systems. Since the transcript has been modified it is not accepted, and actions are automatically taken.
The system protects the employer by one or more of the following:
• Raising a warning the transcript has been modified
• Prohibit accepting the transcript
• Application automatically rejected
• Applicant email automatically returned

Figure 7. Illustrative example of a system protecting the employer in the scenario of a tampered academic credential under the new paradigm.

5.6.3 Example of falsification controls captured in metadata

The code below describes an example of how the metadata of the data product, the academic credentials, accumulates as it is created, shared, modified, and shared again. It maintains a history of who created it, shared it, received it and so on - the chain of custody.

In practice, the metadata that is created and attributed to the academic credentials will adhere to a defined standard. The elements in this metadata are merely examples to demonstrate the concept of how it can accumulate and be acted upon.

<table>
<tr><td><b>1
Academic transcript is created</b></td><td><pre>
metadata {
  doc {
    CredentialID: "12345"
    StudentID: "67890"
    IssuedDate: "2024-12-09"
    Institution: "University of Sydney"
    CredentialType: "Bachelor of Science"
    Status: "Active"
    DigitalSignature: "abc123def456"
    quality: “High”
    format: “pdf”
  }
  m1 {
    created: “2025-01-15 00:00:00”
    created_by: <dgtlid>
    storage: <address>
    access_history: NA
    content_modified: NA
    conditions {
      share: true
      screenshot: true
      modifiable: true
  }
}
</pre></td></tr>
<tr><td><b>2
The transcript is downloaded</b></td><td><pre>
metadata {
  doc {…}
  m1 {…}
  m2 {
    created: “2025-01-15 09:15:04”
    created_by: <dgtlid>
    sent_by: <dgtlid>
    received_by: <dgtlid>
    storage: <address1>
    access_history: <userid1>
    content_modified: NA
    conditions {
      share: true
      screenshot: true
      modifiable: true
  }
}
</pre></td></tr>
<tr><td><b>3
The transcript is modified</b></td><td><pre>
metadata {
  doc {…}
  m1 {…}
  m2 {…}
  m3 {
    created: “2025-01-15 10:45:27”
    created_by: <dgtlid1>
    storage: <address2>
    access_history: <userid1>
    content_modified: true
    conditions {
      share: true
      screenshot: true
      modifiable: true
  }
}
</pre></td></tr>
<tr><td><b>4
The transcript is uploaded to the employer system</b></td><td><pre>
metadata {
  doc {…}
  m1 {…}
  m2 {…}
  m3 {…}
  m4 {
    created: “2025-01-15 11:10:57”
    created_by: <dgtlid>
    sent_by: <dgtlid>
    received_by: <dgtlid>
    storage: <address3>
    access_history: <userid1>
    content_modified: true
    conditions {
</pre></td></tr>
</table>

	share: true screenshot: true modifiable: true } }

5
Employer receives the transcript
	metadata {
	doc {…}
	m1 {…}
	m2 {…}
	m3 {…}
	m4 {…}
	m5 {
	created: “2025-01-15 11:11:07”
	created_by: <dgtlid>
	sent_by: <dgtlid>
	received_by: <dgtlid>
	storage: <address4>
	access_history: <userid1>
	content_modified: true
	conditions {
	share: true
	screenshot: true
	modifiable: true
	}
	}

When the employer receives the academic transcript, the system detects a modification of the document from its original source. The system can then take an action which is specified within the policy of the employer as detailed in Section 5.6.2.

5.6.4 Pseudocode of controls to protect the employer

Below are examples of how protective actions could be activated in the new paradigm. The key innovation is that these controls are generalised, integrated with operating systems or other software, and automatically triggered based on metadata rather than requiring manual implementation.

When an application processes input data, it reads the accompanying metadata to determine appropriate actions based on the organisation’s policies. In this example, the employer’s policy prohibits modified transcripts. When the system detects content modifications in the transcript metadata, it can take one of the following actions outlined above.

In this case the pseudocode details how a warning may be raised and alert the employer via email as well as inform the applicant via email. This may not necessarily be the proposed solution, rather simply demonstrating the action is triggered by the `content_modified` flag in the metadata being `true`.

5.6.4.1 Raise a warning (Guidance)

```
# Define a function to raise a warning with the employer
def raise_warning(transcript, employer_email, applicant_email):
    # Check if the transcript is flagged as modified
    if transcript.is_flagged_as_modified():
        # Compose the warning message
        warning_message = f"""
        Dear Employer,
        The academic transcript submitted by the applicant cannot be trusted as it has been 
flagged for modifications.
        Please request a new, unaltered transcript from the applicant.

        Thank you,
        Credential Verification System
        """

        # Send the warning email to the employer
        send_email(employer_email, "Warning: Untrusted Academic Transcript", 
warning_message)

# Define a function to send an email
def send_email(recipient_email, subject, message):
    # Simulate sending an email
    print(f"Sending email to {recipient_email}")
    print(f"Subject: {subject}")
    print(f"Message: {message}")

# Example usage
class Transcript:
    def __init__(self, data):
        self.data = data

    def is_flagged_as_modified(self):
        # Check metadata for modification flag
        return "modified" in self.data.metadata.content_modified

# Raise a warning if the transcript is flagged as modified
raise_warning(transcript, employer_email="employer@example.com", 
applicant_email="applicant@example.com")
```

5.6.4.2 *Automatically reject the application (Recommendation)*

```
# Define a function to recommend rejecting and filtering out the application
def recommend_reject_application(transcript, application_pool, applicant_email):
    # Check if the transcript is flagged as modified
    if transcript.is_flagged_as_modified():

        # Raise warning to the employer
        recommend_reject_message = f"""
        The academic transcript submitted by the applicant cannot be trusted as it has been 
flagged for modifications. We recommend rejecting the application.

        Would you like to reject the application?
        """

        if ask_reject_application(recommend_reject_message):
                # Remove the application from the pool of legitimate candidates
                application_pool.remove(transcript)

                # Compose the rejection message
                rejection_message = f"""
                Dear Applicant,

                Your application has been rejected because your academic transcript has been 
       flagged for modifications.
                Please submit a new, unaltered transcript to be considered for future 
       opportunities.

                Thank you,
                Credential Verification System
                """

                # Send the rejection email to the applicant
                send_email(applicant_email, "Application Rejected: Untrusted Academic 
       Transcript", rejection_message)
```

```
# Define a function to raise a recommendation with and ask to accept or decline the
recommendation
def ask_reject_application(message):
    result = messagebox.askyesno(message)
    return result

# Define a function to send an email
def send_email(recipient_email, subject, message):
    # Simulate sending an email
    print(f"Sending email to {recipient_email}")
    print(f"Subject: {subject}")
    print(f"Message: {message}")

# Example usage
class Transcript:
    def __init__(self, data, metadata):
        self.data = data

    def is_flagged_as_modified(self):
        # Check metadata for modification flag
        return "modified" in self.metadata.content_modified

# Example application pool
application_pool = [transcript]

# Reject the application if the transcript is flagged as modified
recommend_reject_application(transcript, application_pool,
applicant_email="applicant@example.com")
```

5.6.4.3 Application is prohibited from being submitted (Restriction)

```
# Define a function to check if the transcript is flagged as modified
def is_transcript_modified(transcript):
    # Check metadata for modification flag
    return "modified" in transcript.metadata.content_modified


# Define a function to handle the application submission
def handle_application_submission(transcript, portal):
    # Check if the transcript is flagged as modified
    if is_transcript_modified(transcript):
        # Disable the submit button on the portal
        portal.disable_submit_button()

        # Display a message to the applicant
        portal.display_message("Your academic transcript has been flagged for modifications
and cannot be accepted. Please submit a new, unaltered transcript.")

# Define a class for the Portal
class Portal:
    def __init__(self):
        self.submit_button_active = True

    def disable_submit_button(self):
        self.submit_button_active = False
        print("Submit button is now inactive.")

    def display_message(self, message):
        print(f"Portal Message: {message}")

# Example usage
class Transcript:
    def __init__(self, data):
        self.data = data

# Create a portal instance
portal = Portal()
```

```
# Handle the application submission
handle_application_submission(transcript, portal)
```

5.6.4.4 Applicants email is automatically returned (Restriction)

```
# Define a function to return the applicant's email with reasons and steps to resolve
def return_email_with_resolution(transcript, applicant_email):
    # Check if the transcript is flagged as modified
    if transcript.is_flagged_as_modified():
        # Compose the return message with reasons and steps to resolve
        return_message = f"""
        Dear Applicant,

        Your academic transcript has been flagged for modifications and cannot be accepted.
        Reason: The transcript shows signs of alteration.

        Steps to resolve:
        1. Obtain a new, unaltered transcript from your educational institution.
        2. Ensure that the new transcript is digitally signed and verified by the
institution.
        3. Resubmit the new transcript through the credential-sharing platform.

        Thank you,
        Credential Verification System
        """

        # Send the return email to the applicant
        send_email(applicant_email, "Action Required: Submit New Transcript",
return_message)

# Define a function to send an email
def send_email(recipient_email, subject, message):
    # Simulate sending an email
    print(f"Sending email to {recipient_email}")
    print(f"Subject: {subject}")
    print(f"Message: {message}")

# Example usage
class Transcript:
    def __init__(self, data):
        self.data = data

    def is_flagged_as_modified(self):
        # Check metadata for modification flag
        return "modified" in self.metadata.content_modified


# Return the applicant's email with reasons and steps to resolve if the transcript is
flagged as modified
return_email_with_resolution(transcript, applicant_email="applicant@example.com")
```

6 Sensitivity Calculus

Tabular data sharing presents unique challenges and opportunities in data protection frameworks. While we've examined document formats like PDFs for academic transcripts, structured datasets organised in rows and columns (spreadsheets or database tables) require different handling considerations.

A critical practice in data sharing involves joining and linking tables from different sources using common identifiers. For example, customer information might be linked with purchase history via customer IDs, creating comprehensive, unified views for analysis.

Data sharing offers substantial benefits:

1. **Enhanced Collaboration**: Enables effective partnerships across researchers and organisations that drive innovation, leading to innovative solutions and advancements in various fields.

2. **Resource Optimisation**: Reduces duplication of efforts and increasing operational efficiency.

3. **Accelerated Research**: Shortens research timelines by leveraging existing datasets.

4. **Informed Decision-Making**: Provides broader perspectives through access to diverse data sources.

However, joining datasets introduces specific privacy risks. To address these concerns, we introduce sensitivity calculus—a framework specifically designed to prevent the combination of separate pieces of personal information that could result in personally identifiable information.

This framework provides systematic methods for evaluating when and how data combinations might elevate privacy risks, allowing organisations to implement appropriate controls before sensitive data exposures occur.

6.1 Overview

Sensitivity calculus serves as a decision-making framework governing the combination of multiple data products through systematic evaluation of their sensitivity and compliance levels. This approach ensures data aggregation remains within permissible sensitivity thresholds and adheres to contextual usage rules.

Consider personal information like date of birth and stress address. Each may be acceptable to include in a data product individually, but when combined, they create a higher sensitivity level that may exceed established constraints. In such cases, the system protects users by either preventing data sharing or issuing clear risk warnings.

The framework operates through three core mechanisms:

- **Metadata-driven decisions**: Each data product carries metadata encoding its sensitivity level, usage restrictions, and contextual rules, enabling automated decision-making.
- **Combination rules**: When data products merge, the system applies appropriate rules—whether additive sensitivity calculations (combining "Low" with "Medium"

creates "High"), restrictive policies (prohibiting combinations with health data), or hierarchical overrides (where "Critical" classification dominates other ratings).

- **Compliance assurance**: The system compares the calculated combined sensitivity against a predefined threshold to determine whether the proposed data operation should proceed.

6.2 Pseudocode for Sensitivity Calculus

The pseudocode below is an example of who the logic may operate within the new system.

```
Input Data:
  - A set of Data Products: {DataProduct1, DataProduct2, ..., DataProductN}
  - Metadata for each Data Product, including:
    - SensitivityLevel: Ordinal value (e.g., Low, Medium, High, Critical)
    - CombinationRule: Governs sensitivity interaction (e.g., Additive, Maximum, Restrictive)
    - ContextualRestrictions: Defines prohibited combinations (e.g., "Cannot combine with personal health data")
  - TargetSensitivity: The allowable sensitivity threshold for the combined data.
c
Initialise:
  - Set CombinedSensitivity to the lowest possible level (e.g., "Low").
  - Create an empty list of Violations for tracking rule conflicts.

Process Each Data Product:
  - For each DataProduct in the set:
    - Retrieve SensitivityLevel, CombinationRule, and ContextualRestrictions from Metadata.
    - Apply CombinationRule:
      - If "Additive": Add SensitivityLevel to CombinedSensitivity.
      - If "Maximum": Update CombinedSensitivity to the higher value between CombinedSensitivity and SensitivityLevel.
      - If "Restrictive": Set CombinedSensitivity to the most restrictive value (e.g., "Critical").
    - Check ContextualRestrictions:
      - If the combination violates a restriction (e.g., "Cannot combine with sensitive financial data"), record the conflict in Violations.

Evaluate Combined Sensitivity:
  - If CombinedSensitivity ≤ TargetSensitivity and Violations is empty:
      - Approve the operation.
  - Else:
      - Deny the operation and provide reasons based on Violations and sensitivity thresholds.

Output Results:
  - Decision: Approved or Denied.
  - Explanation: A detailed summary of the sensitivity analysis and any violations detected.
```

6.3 Example Scenario: Academic and performance data integration

An employer seeks to combine an applicant's academic transcript (sensitivity: Medium) with a performance evaluation from a previous employer (sensitivity: High). The organisation's acceptable combined sensitivity threshold for this operation is set at 'High' sensitivity.

The metadata for these documents contains specific combination rules:

- The transcript must not be combined with personal health data.
- The performance evaluation automatically escalates to ‘Critical’ sensitivity if paired with identifiable financial records.

Automated evaluation process:

1. **Sensitivity calculation:**
    - System compares "Medium" (transcript) with "High" (evaluation)
    - Applying the "Maximum" rule yields a CombinedSensitivity of "High"
    - System verifies no contextual restrictions are violated.
2. **Compliance verification:**
    - The CombinedSensitivity ("High") meets the TargetSensitivity ("High")
    - No rule violations detected in the proposed combination.
3. **Result:**
    - Operation is approved
    - Combined data is securely shared with appropriate controls.

7 Conclusion

The new paradigm of data sharing fundamentally transforms data product management and protection throughout the entire lifecycle. As data products move through creation, collection, storage, modification, and sharing, their metadata creates a comprehensive digital fingerprint. This accumulated record—capturing quality metrics, format specifications, collection conditions, access history, and transformations—enables automated risk assessment and decision-making without human intervention.

At the core of this paradigm shift is the data product. It initiates and enforces protection rules within a system. By transferring control from systems and maintainers to the data product, we ensure protective controls remain continuously active across all environments in which the data product resides. This approach dramatically reduces reliance on manual intervention and user judgement, making data sharing both more efficient and inherently secure. In essence, the system protects the user.

The ability to automatically interpret metadata and enforce machine-readable policies represents a significant advancement in data protection. Through real-time risk detection and mitigation, the system prevents misuse while preserving data integrity. This automated protection not only alleviates cognitive and administrative burdens but fundamentally strengthens trust in shared data authenticity, reliability, and reduced exposure to risks. The result is an ecosystem where the system protects the user.

Disclaimer: This document comprises and is the subject of intellectual property (including copyright) and confidentiality rights of one or multiple owners and where applicable, its third-party data owners (Data Providers), together (IP Owners). The information contained in this document may have been prepared using raw data owned by the Data Providers. The Data Providers have not been involved in the analysis of the raw data, the preparation of, or the information contained in the document. The IP Owners do not make any representation (express or implied), nor give any guarantee or warranty in relation to the accuracy, completeness or appropriateness of the raw data, nor the analysis contained in this document. None of the IP Owners will have any liability for any use or disclosure by the recipient of any information contained in, or derived from this document. To the maximum extent permitted by law, the IP Owners expressly disclaim, take no responsibility for and have no liability for the preparation, contents, accuracy or completeness of this document, nor the analysis on which it is based. This document is provided in confidence, may only be used for the purpose provided, and may not be copied, reproduced, distributed, disclosed or made available to a third party in any way.

UNSW SYDNEY | UTS

Trustworthy Digital Society